

## Приложение Fitmus version 2.

### 1. Разработка приложения

#### 1.1 Технические аспекты

- Платформа: phonegap + jquerymobile. На выходе ожидается код для генерации приложений в облаке phonegap. Генерация приложений осуществляется силами заказчика.
- Минимальное разрешение экрана 320x480
- Максимальное разрешение экрана 1920x1080, приложение должно быть «резиновым», пропорции блоков сохраняются в соответствии с макетами, кроме изображений упражнений – они имеют фиксированный размер (160x160px).
- Приложение является тонким клиентом сервиса Fitmus.com
- Мобильное приложение будет общаться с сервером по API (сервер сайта fitmus.com).
- API регистр зависимо – все ключи массивов/параметры передавать в нижнем регистре, если не указано иное отдельно.
- **Технические аспекты, как и все детали приложения – обсуждаемы, так как вы специалист в разработке– то с Вашей стороны ожидаются советы\рекомендации.**
- На усмотрение разработчика возможно изменение дизайна приложения (в разумных пределах)
- Приложение должно уметь работать в offline режиме, для этого оно должно уметь хранить данные пользователя внутри мобильного устройства.

#### 1.2 Основная цель приложения

Приложение ориентировано на людей, занимающихся в спортзале и которым необходимо вести дневник своих тренировок – т.е. видеть какие упражнения выполняются в тот или иной день, а также иметь возможность вводить данные о выполненных упражнениях – сколько раз и с каким весом выполнил упражнение и прочие данные. Данное приложение будет дополнением к сервису fitmus.com.

Основная техническая задача мобильного приложения – быть посредником для данных между пользователем и сервером. Существенной обработки/расчета данных - в мобильном приложении не будет.

Приложение должно уметь работать offline, т.е. без постоянного доступа к серверу (кроме первого запуска, после установки приложения), т.е. по сути, краткая схема такая:

1. Синхронизация данных с сервером (ручная или автоматическая)
2. Отображаем данные пользователю
3. Получаем данные по тренировке от пользователя
4. Сохраняем данные пользователя в мобильном устройстве до следующей синхронизации (ручной или автоматической). Переход на пункт 1.

**Примечание:** сейчас в google play и appstore доступна первая версия приложения, которую можно установить и посмотреть в обще-ознакомительных целях. Так же рекомендуется посмотреть сайт fitmus.com.

### 1.3. Взаимодействие с сервером

Так как приложение должно будет уметь работать offline, то необходимо синхронизировать данные приложения и сервера. Синхронизация будет двух типов:

1. **Синхронизация вспомогательных данных** – списки мышц, упражнений, изображения упражнений, параметры упражнений и прочая вспомогательная информация.

Выполняется обязательно:

- при первом запуске приложения после установки и авторизации пользователя
- при смене аккаунта пользователя (т.е. пользователь разлогинился и залогинился снова), так как у них могут быть разные данные, в том числе и списки упражнений
- при каждом запуске приложения, если в настройках стоит «Автоматическая синхронизация» и доступен интернет
- при ручной синхронизации

2. **Синхронизация пользовательских данных** – результаты тренировок и текстовые записи.

Выполняется обязательно - во всех случаях, что и синхронизация вспомогательных данных + при выходе из приложения, если в настройках стоит «Автоматическая синхронизация».

**Нам видится это так:**

- при авторизации пользователя создается некое хранилище только для данного пользователя
- Затем происходит синхронизация данных, в ходе которых приложение получает много вспомогательной информации (наполняется данными), которая сохраняется исключительно в хранилище авторизованного пользователя.
- Далее в ходе работы приложения – пользователь вносит информацию, которая так же сохраняется в хранилище.
- В момент синхронизации (ручной или автоматической) приложение отправляет все измененные пользовательские данные на сервер.
- При смене аккаунта хранилище пользователя не должно удаляться, так как там могут остаться не синхронизированные с сервером пользовательские данные

#### **ПРИМЕЧАНИЕ:**

Все данные пользователя (тренировки\записи) группированы по дням. При синхронизации с сервером надо отправлять на сервер данные только по тем дням, где что-либо менялось. Данные по дням, в которых ничего не менялось – посылать на сервер не надо!

Предлагается по ходу работы приложения сразу готовить данные для сервера (дублирование данных) или маркировать неким образом измененные дни. Причем после синхронизации – маркировка или дубли должны сбрасываться, т.е. повторная синхронизация (без изменения данных) ничего не должна отправить на сервер.

Новые данные, пришедшие из приложения при синхронизации, перезапишут данные на сервере и сервер вернет уже ВСЕ полные данные, вместе с изменениями.

По сути, приложение всегда отправляет только изменения, сервер их применяет и всегда отдает полный ответ, который приложения должно сохранить у себя вместо старых данных. Отмечаем - в пришедших от сервера данных могут быть данные отсутствующие в приложении, потому ответ от сервера должен полностью заменять данные в приложении.

#### 1.4. Условия

##### От Вас

От Вас ожидается конечный готовый и работоспособный продукт, готовый к загрузке в облако phonegap для генерации приложений. Сама генерация приложений осуществляется силами заказчика.

Весь интерфейс должен быть реализован через шаблоны, код должен быть читаемым, и комментирован по мере необходимости – т.е. нужен код, который можно будет потом поддерживать и развивать.

##### От нас

Необходимое API, в том числе доработки и все что касается серверной стороны + материалы (иконки, картинки, тексты).

При необходимости можем изменить API под Ваши требования – это все обсуждаемо.

## 2. Описание экранов

### 2.1 Авторизация



Если пользователь был авторизован ранее, то стартовым экраном является «2.2 Главный экран».

Если пользователь не авторизован в приложении, то показывается данный экран. После ввода данных, они передаются на сервер через API и если данные верные – приложение запоминает пользователя, иначе показывает ошибку, текст которой получен от сервера API.

«О проекте» и «Регистрация» - ссылки, при клике на которые должен открываться стандартный браузер пользователя и отображать следующие url (если можно ссылки открыть внутри приложения – значит делаем внутри приложения):

«О проекте» - <http://fitmus.com>

«Регистрация» - <http://fitmus.com/auth/registration>

**Примечание:** до ввода данных в поля, там отображаются дефолтные надписи – в поле логин «Логин», а в поле пароль – «Пароль». При клике на поле – надпись соответствующего поля исчезает, и дается возможность пользователю ввести данные.

После авторизации, если данных пользователя нет (первая авторизация в приложении), то необходимо насильно запустить «Синхронизацию».

## 2.2 Главный экран



После авторизации пользователь попадает на данный экран.

Здесь отображается список упражнений из расписания пользователя на выбранный день (по умолчанию открывается сегодняшний день).

1. Кнопка запускает расширенный режим (отображаются дополнительные блоки в верстке см. «2.3 Главный экран – расширенный режим»)

2. Кнопка открывает календарь (jquery mobile datepicker). При выборе конкретного дня – переходить на выбранный день

3. Кнопка открывает главное меню приложения – экран **2.7 Меню**

4. Название дня недели и дата открытого дня, по умолчанию это текущий календарный день (сегодня).

5. Кнопка перейти на день назад (вчера)

6. Кнопка перейти на день вперед (завтра)

7. Картинка упражнения, у всех упражнений они могут быть разные. Картинки грузятся с сервера в момент синхронизации и сохраняются на телефоне пользователя. Размер фиксированный – 160x160 пикселей.

8. Название упражнения. Цвет зависит от поля mode(см. АПИ).

Планируется 4 варианта цвета по степеням нагрузки по данному упражнению – тяжелая, средняя, легкая, нормальная.

9. Группа мышц, для которой планируется упражнение.

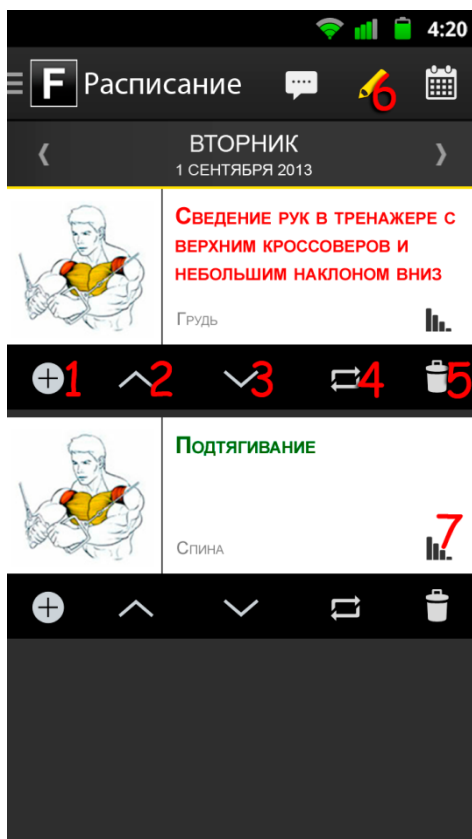
10. Кнопка комментариев – при клике появляется всплывающее окно, на почти весь размер экрана с комментариями по всем упражнениям из текущего дня. То есть берутся комментарии и сливаются в один день (переводы строк сохраняются). Данные комментарии только для чтения. Смотри экран «**2.10 Комментарии**»

11. Кнопка изменения уровня тяжести тренировки. Тяжесть может быть – нормальной, тяжелой, средней, легкой. Данная кнопка меняет тяжесть упражнения по кругу. При смене тяжести – меняется и цвет названия упражнения.

При клике на **7,8,9**, т.е. при клике на упражнение (на всю строку упражнения, кроме кнопки 11), открывается экран «**2.6 Упражнение**»

Если для страницы можно применить горизонтальный свайп(swipe) с дублированием функционала кнопок 5 и 6 (листание дней вперед\назад) – то значит применяем (желательный функционал).

## 2.3 Главный экран – расширенный режим



В расширенном режиме становятся видны дополнительные блоки с инструментами, относящиеся к своим упражнениям. В данном режиме, кнопка **6**, становится желтой. Отступ между упражнениями (строками) несколько увеличивается (на усмотрение разработчика).

**Примечание:** все упражнения имеют поле position, которое указывает порядок выполнения упражнений, т.е. эдакий порядковый номер упражнения.

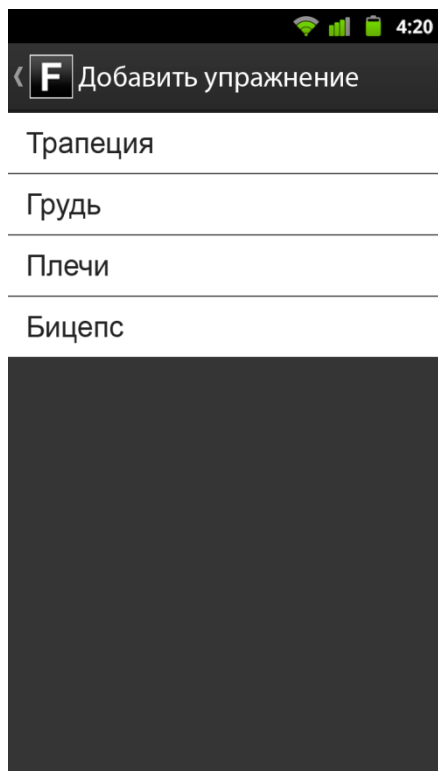
1. Кнопка добавить упражнение в выбранный день см. экран «2.4 Выбор мышцы». Новое упражнение добавляется после упражнения, в котором нажали на данную кнопку, все упражнения, что ниже, естественно меняют position, так как они сдвинулись вниз.
2. Кнопка поднять упражнение вверх - данная кнопка поднимает упражнение вверх, меняя position у себя и у верхнего упражнения (обмен местами)
3. Кнопка опустить упражнение вниз, аналогично поднятию, только вниз.
4. Кнопка заменить упражнение. По сути - это удаление текущего упражнения и добавление нового упражнения на место удаленного. Новому упражнению передается position старого упражнения.

5. Удаление упражнения. Все упражнения что ниже – меняют position.
6. Кнопка возврата в стандартный режим отображения
7. Кнопка изменения уровня тяжести тренировки. Тяжесть может быть – нормальной, тяжелой, средней, легкой. Данная кнопка меняет тяжесть упражнения по кругу. При смене тяжести – меняется и цвет названия упражнения. В режиме редактирования данная иконка становится ярче.

Т.е. суть проста – на выходе надо получить список упражнений, с уникальными position, которые покажут очередность выполнения упражнений.

Повторный клик по желтой кнопке **6 (карандаш)** отключает расширенный режим – экран «2.2 Главный экран»

## 2.4 Выбор мышцы



При добавлении упражнения, сначала необходимо выбрать мышцу, для которой будут показаны упражнения. Список мышц берется через АПИ при синхронизации и сохраняется в приложении. После выбора мышцы переходим на экран «**2.5 Выбор упражнения**»

При нажатии на кнопку в баре «Добавить упражнение» - возвращаемся назад на «**2.3 Главный экран – расширенный режим**»

## 2.5 Выбор упражнения



На данном экране отображается список упражнений, для выбранной мышцы. То есть – для каждой мышцы свой набор упражнений. Список мышц и упражнений для них берется через АПИ и сохраняется в приложении.

Добавление упражнения в расписание происходит через нажатие на кнопку «+». После нажатия на «+» сразу возвращаемся в экран «**2.3 Главный экран – расширенный режим**», с уже добавленным новым упражнением.

При нажатии на кнопку в баре «Добавить упражнение» - возвращаемся назад на «**2.4 Выбор мышцы**»



## 2.6 Упражнение



Экран упражнений может быть двух видов, в зависимости от настроек, а именно поля «Отображать время» в настройках.

1. Ссылка на упражнение, пример - <http://fitmus.com/exercise/id/1> . По клику открывается браузер пользователя и загружается данная страница. Если можно открывать url прямо в приложении – открываем.

2. Ссылка на мышцу упражнения, пример - <http://fitmus.com/muscle/group/7> . Если можно открывать url прямо в приложении – открываем.

3. Поле для ввода результата подхода. Может принимать как целые, так и дробные числа, в зависимости от типа поля (см. API). Точность отображения (именно отображения) данных – до 2 знаков после точки. Разделитель дробной части – точка.

4. Отображение данных с предыдущей тренировки, где было использовано данное упражнение (поиск предыдущего упражнения осуществляется в самом приложении). Если данных по предыдущему упражнению нет, то ставим прочерк.

5. Единицы измерения, могут быть следующих видов:
- Вес (килограммы, фунты)
  - Расстояния (метры, километры, мили, футы)
  - Разы

Единицы измерений и их коэффициенты передаются с сервера.

В рамках одного параметра можно переключаться между единицами измерений по кругу, к примеру – кг -> фунты -> кг и т.д. При переключении меняется значение полей номер 3 и 4. Т.е. если введено 95 кг и пользователь нажимает на кнопку 5 (кг), то все веса в рамках текущего подхода переводятся в следующую систему, в данном случае – фунты, и станет 209.44 фунта. Данная система уже реализована на сайте fitmus.com и можно посмотреть там.

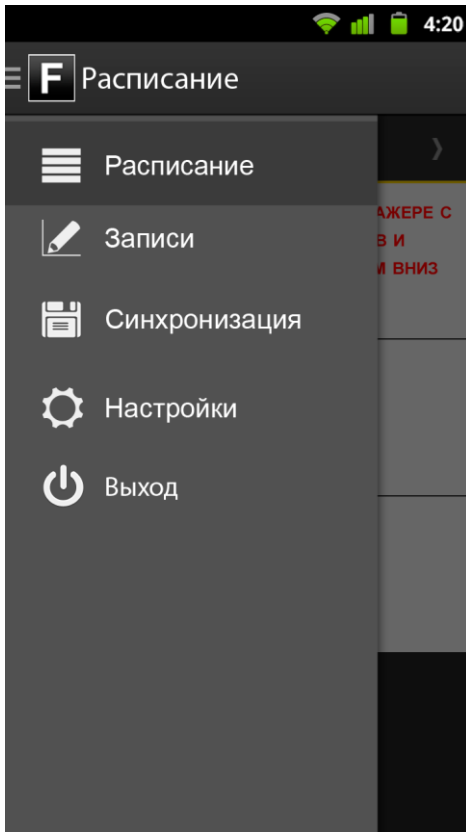
6. Кнопка таймера. Т.е. когда пользователь нажимает на данную кнопку, то в поле, к которому привязана данная кнопка начинает идти отсчет от 00:00 до 59:59. При повторном нажатии – таймер останавливается. Таймер срабатывает, только если поле пустое или имеет значение 00:00. Если кликнуть по самому полю времени, то можно установить время вручную через jquery DurationFlipBox.

При достижении значения 59:59 таймер автоматически останавливается.

### Особенности:

- если работает таймер «отдых» и нажимают по таймеру «работа» в том же подходе\строке, который имеет пустое значение или 00:00, то работающий таймер «отдых» останавливается, а таймер «работа» стартует отсчет. Если значение в поле «работа» не пустое – то нажатие никак не обрабатывается, ничего не меняется.
  - Если работает таймер «работа» и нажимают на таймер «отдых» в следующем подходе, который имеет пустое значение или 00:00, то работающий таймер «работа» останавливается и стартует таймер «отдых» в следующем подходе (по которому кликнули). Если значение в поле «отдых» не пустое – то нажатие никак не обрабатывается, ничего не меняется.
  - Вся эта логика нужна упрощения процедуры – останови один таймер, запусти следующий – т.е. делать это в один клик, а не в два.
7. Время отдыха, измеряется в минуты : секунды. При клике по полю использовать jquery DateBox (<http://dev.jtsage.com/jQM-DateBox/>), в частности DurationFlipBox.
  8. Время выполнения упражнения, при клике по полю использовать jquery DateBox.
  9. Удаление подхода. Все остальные подходы, что находятся ниже – сдвигаются вверх. Т.е. пропусков быть не должно. Перед удалением – должно быть диалоговое окно с подтверждением действия - «Точно удалить?» и варианты ответа «да\нет». Порядковые номера подходов так же сдвигаются – перепрыгиваний в номерах быть не должно.
  10. При клике на баре «Вторник, 1 сентября» - возвращаться назад к списку упражнений по выбранному дню.
  11. Кнопка открывает рорир и показывает комментарии к текущему упражнению. Комментарии только для чтения. Смотри экран **«2.10 Комментарии»**
  12. Кнопка изменения уровня тяжести тренировки. Тяжесть может быть – нормальной, тяжелой, средней, легкой. Данная кнопка меняет тяжесть упражнения по кругу. При смене тяжести – меняется и цвет названия упражнения.

## 2.7 Меню



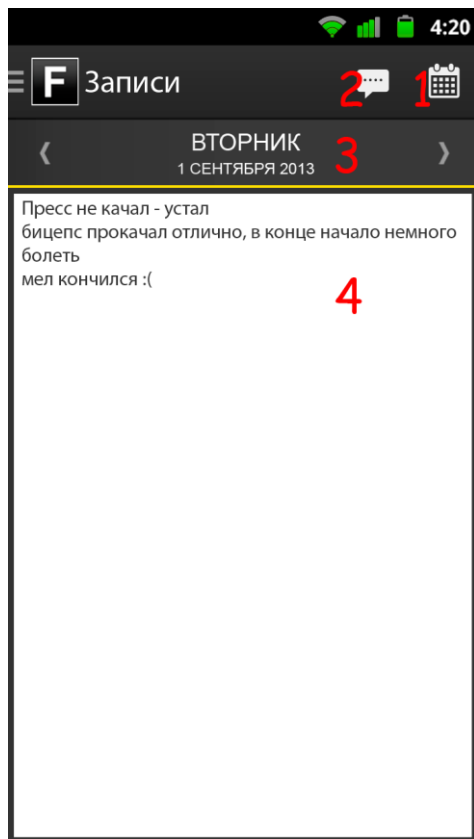
- Расписание – экран «**2.2 Главный экран**»
- Записи – экран «**2.8 Записи**»
- Синхронизация – экран не меняется, а просто выполняется полная синхронизация приложения, при этом должна быть любая анимация «ожидания», которая не даст пользоваться приложением во время синхронизации. Анимация – на усмотрение разработчика.

Перед синхронизацией надо выполнить проверку наличия интернета, и если соединения нет – то показать соответствующую ошибку «Нет соединения с интернетом».

В случае если соединение с интернетом есть, но сервер не доступен, то писать «Сервер не доступен – технические работы».

- Настройки – экран «**2.9 Настройки**»
- Выход – приложение закрывается, при этом авторизация пользователя сохраняется и при последующем запуске экран авторизации НЕ появляется.

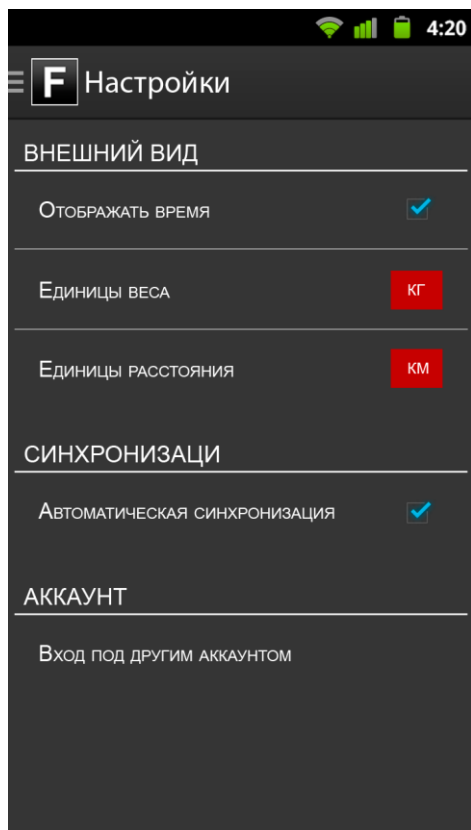
## 2.8 Записи



Здесь пользователь читает\редактирует записи по ходу тренировки.

1. Календарь, аналогичен тому, что в экране «2.2 Главный экран» - т.е. выбирает день, записи которого надо отобразить.
2. Комментарии, аналогичны тем, что в экране «2.2 Главный экран» - т.е. показывает комментарии всех упражнений к тренировке на данный день.
3. Выбор даты и дата, аналогичны тем, что в экране «2.2 Главный экран» - т.е. переключает дни вперед\назад и показывает выбранную дату. По умолчанию – открывается текущий день (сегодня).
4. Текст на выбранный день, полученный с сервера и ДОСТУПНЫЙ для редактирования. Должно быть поле textarea. По высоте растягивать на весь экран, НО предусмотреть возможность прокручивать текст вниз\вверх, так как текст может не влезть на один экран, т.е. предусмотреть скролл\свайп.

## 2.9 Настройки



«**Отображать время**» - определяет вид экрана «2.6 Упражнение» - либо экран, расширенный со временем\таймерами, либо сжатый без времени\таймеров.

«**Единицы веса**» - единицы по умолчанию отображающиеся для ввода веса

«**Единицы расстояния**» - единицы по умолчанию отображающиеся для ввода расстояния.

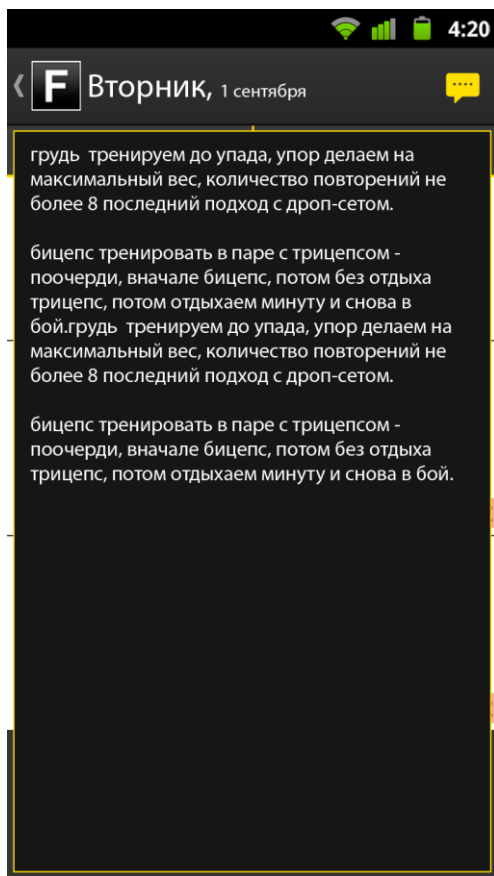
При клике по единицам – они меняются, как и цвет фона единиц. Списки единиц, цветов и коэффициентов берутся по API с сервера.

С сервера всегда приходят данные в килограммах\метрах. Если тут (в настройках) выбираются другие единицы измерений, то они отображаются как стартовые единицы измерений на всех экранах приложения, а данные преобразуются из килограмм\метров в выбранные единицы измерений.

«**Автоматическая синхронизация**» - автоматически запускается синхронизация при входе в приложение и выходе из него.

«**Вход под другим аккаунтом**». Данные авторизации текущего пользователя стираются (токен, пароль), но его данные по тренировкам и записям остаются сохраненными на мобильном устройстве, так как там могут быть не синхронизованные с сервером данные. Далее приложение показывает экран «**2.1 Авторизация**». Т.е. реализуется некий многопользовательский режим.

## 2.10 Комментарии



При нажатии на значок комментария (находится в верхнем баре на экранах «2.6 Упражнение», «2.2 Главный экран», «2.3 Главный экран- расширенный режим») – значок загорается желтым цветом и появляется всплывающее окно (popup) с комментариями по текущей тренировке. Повторный клик по иконке – скрывает окно комментариев.

Текст комментариев нельзя редактировать – т.е. он в режиме только чтения.

Предусмотреть возможность пролистывания текста вниз, так как он может не поместиться на одном экранном листе. Использовать для этого вертикальный свайп (swipe) или скролл.

По сути данный функционал – быстрый доступ к информации по тренировке текущего дня или текущего упражнения.

## Подробное описание части экрана упражнения



На данном изображении показана часть экрана «Упражнение» (10-ый подход в упражнении) и его изменения при клике на единицы измерения.

Клики на единицах измерения показаны темными кружочками, а изменяющиеся части экрана – сиреневыми. В данном случае кликаем только по одному изменяемому параметру. Шаги такие:

1. Изначально: текущий результат 100метров и прошлый результат (было) 200 метров
2. После клика метры превращаются в километры, причем как текущий результат, так и прошлый результат переводятся в километры.
3. Еще клик и следующая система измерений – мили. Опять переводим и текущий результат и прошлый.
4. Очередной клик перебрасывает в начало единиц измерений – на метры, так как список единиц измерений расстояний у нас кончился.

Список единиц измерений берется с сервера при синхронизации.

### Примечание:

- у каждой единицы измерений свой цвет кнопки, который также задается сервером
- данные надо отображать в поле ввода с точностью до 2 знаков после точки, однако многократное изменение единицы измерения не должно влиять на конечный результат. К примеру - если я ввел 100 метров и 100 раз переконвертировал единицы измерения – в итоге должно получиться ровно 100 метров (на сайте это реализовано – можно посмотреть как).
- Список единиц измерений и их коэффициентов для перевода друг в друга берется с сервера при синхронизации.

## Отображение уведомлений и диалоговых окон (popup)

Использовать инструменты jquery с темной темой оформления, близкой к цветам приложения. Остальное на усмотрение разработчика.



## 1. Основы взаимодействия с АПИ

### Описание параметров запросов

**app\_id** – идентификатор приложения, выдается заказчиком, **обязателен во всех запросах** к АПИ

**app\_key** – ключ приложения, выдается заказчиком и никогда не передается в открытом виде. Используется для подписи приложения и данных.

**app\_sign** – подпись приложения, генерируется в приложении, **обязательна во всех запросах** к АПИ

Формула генерации: `md5(app_id + app_key)`

**user\_id** – идентификатор пользователя приложения, передается везде, кроме запроса авторизации

**token** – токен (подпись) пользователя, выдается сервером при авторизации пользователя и используется во всех запросах, где передается `user_id`. Без подписи `user_id` не принимается. При авторизации пользователя надо сохранять `token` в приложении до момента разлогинивания.

### Пример:

GET

[http://fitmus.com/api/v3/train?app\\_id=1&app\\_sign=098f6bcd4621d373cade4e832627b4f6&user\\_id=1&token=e11cbb19052e40b07aac0ca060c23ee](http://fitmus.com/api/v3/train?app_id=1&app_sign=098f6bcd4621d373cade4e832627b4f6&user_id=1&token=e11cbb19052e40b07aac0ca060c23ee)

### Форматы ответа

Ответ сервера имеет жесткую структуру двух видов – успешный ответ и ответ с ошибкой. Формат ответа - json.

#### Успешный:

```
{
  info: <string>[]      // массив строк для вывода пользователю в виде сообщений рорир
  data: <mixed>         // массив данных
}
```

#### Ошибка:

```
{
  info: <string>[]      // массив строк для вывода пользователю в виде сообщений рорир
  error: {              // описание самой ошибки
    code:<int>           // код ошибки
    message:<string>     // сообщение ошибки – выводить пользователю в рорир
  }
  input:{               // для debug – те данные что пришли на сервер
    method:<string>      // метод запроса
    post: <mixed>        // данные из POST массива
    get: <mixed>         // данные из GET массива
  }
}
```

#### Примечание:

- далее в данной документации ответ будет показан в виде только поля **data**, для экономии места.
- Параметр `info` – возможность сервера передать пользователю какие либо мелкие извещения, к примеру - о необходимости «сменить пароль» или о «выходе обновления приложения».

## 2. Авторизация пользователя

**GET /api/v3/login**

### **Input:**

login – логин или email пользователя, сервер сам определит что передано (регистр сохранять)  
passwd – пароль пользователя (регистр сохранять)

### **Output:**

```
data: {  
    id:<int>           // идентификатор пользователя – нужен во всех остальных запросах  
    name: <string>     // полное имя пользователя  
    login: <string>    // логин пользователя  
    token: <string>    // токен пользователя – нужен во всех остальных запросах  
}
```

### 3. Получение общих данных пользователя

GET /api/v3/syncdata

#### Input:

user\_id – идентификатор пользователя

token – токен пользователя

#### Output:

```
data: {
  units: <Units> // единицы измерений и коэффициенты
  mode: { // возможные значения тяжести тренировки
    <string>: { // техническое название тяжести – Normal/Hard/Medium/Easy
      id:<int> // идентификатор тяжести (порядковый номер)
      name:<string> // техническое название тяжести – Normal/Hard/Medium/Easy
      view:<string> // название тяжести для пользователя - русифицированное
      color:<string> // цвет нагрузки (для названия упражнения) пример - #007FCC
    },
    ... // таких записей много - Normal/Hard/Medium/Easy
  }
  musclegroup: {
    <int>:<string>, // id мышечной группы : ее название
    ... // таких записей может быть много – для всех мышечных групп
  },
  musclegroup_exercise: { // тут показаны - какие упражнения для каких мышц
    <int>: <int>[], // id мышечной группы : массив id упражнений для данной мышцы
    ... // таких записей может быть много – для всех мышечных групп
  },
  exercise: {
    <int>: { // идентификатор упражнения
      id:<int> // идентификатор упражнения (дубль)
      name:<string> // название упражнения
      img:<string> // полный путь к картинке упражнения
      type:<string> // тип параметров для ввода результат (w/c/d/t)
    },
    ...
  }
}
```

#### Примечание:

w/c/d/t – это сокращения от weight/count/distance/time. То есть это основные параметры результатов упражнения – вес, количество повторений, дистанция (бег, плавание), время упражнения. На данный момент есть следующие комбинации: wc (вес на количество повторений) и dt(расстояние на время преодоления расстояния).

**Пример:** если в поле type пришло значение «wc», то это значит что надо вывести два поля – для ввода веса и количество повторений. По умолчанию единицы измерения – килограммы, разы, метры, минуты.

## 4. Получение данных о тренировках

GET /api/v3/train

### Input:

user\_id – идентификатор пользователя для которого получаем тренировку  
token – токен этого пользователя

### Output:

```
data: {  
    <timestamp>: <Exercise>[]           // дата тренировки: массив упражнений  
}
```

### Примечание:

Результат выполнения упражнения содержит сокращения – w/c/d/t – weight/count/distance/time.

Могут быть еще два дополнительных параметра:

**tr** - time recreation – время отдыха

**tw** – time work – время работы

Данные **tr** и **tw** могут быть не всегда, а только там где пользователь их ввел. Измеряются\хранятся в секундах, но отображаются пользователю в виде мм:сс (минуты:секунды).

**Пример результата** первого подхода - было жим штанги 12 раз по 100кг, перед упражнением отдыхали 3 минуты и 20 секунд, выполнили упражнение за 21 секунду:

```
"result": {                                     // результат тренировки  
    "1": {                                     // первый подход  
        "c": 12,                             // совершено 12 повторов (count)  
        "w": 100                             // вес (weight) снаряда – 100 кг  
        "tr": 200                             // отдыхали перед упражнением 200 секунд – 3:20  
        "tw": 21                             // работали 21 секунду – 0:21  
    },  
    ...  
}
```

### Пример:

```
"data": {
  "1379145600": [
    {
      "date": "2013-09-14 12:00:00",
      "date_unix": "1379145600",
      "id_exercise": "12",
      "id_muscle_group": "2",
      "mode": "Hard",
      "position": "0",
      "result": {
        "1": {
          "c": 12,
          "w": 12
        },
        "2": {
          "c": 12,
          "w": 13
        }
      },
      "status": "Active",
      "comment": "Тест<br>тест"
    },
    {
      "date": "2013-09-14 12:00:00",
      "date_unix": "1379145600",
      "id_exercise": "7",
      "id_muscle_group": "2",
      "mode": "Hard",
      "position": "1",
      "result": {
        "1": {
          "c": 15,
          "w": 15
        },
        "2": {
          "c": 16,
          "w": 16
        }
      },
      "status": "Active",
      "comment": " другой комментарий"
    }
  ],
  "1379318400": [
    {
      "date": "2013-09-16 12:00:00",
      "date_unix": "1379318400",
      "id_exercise": "39",
      "id_muscle_group": "6",
      "mode": "Easy",
      "position": "0",
      "result": null,
      "status": "Active",
      "comment": ""
    }
  ]
}
```

// дата тренировки в timestamp  
// начало упражнения №1  
// дата тренировки  
// дата тренировки в timestamp  
// идентификатор упражнения  
// идентификатор мышечной группы  
// тяжесть тренировки - тяжелая  
// позиция упражнения – делаем его первым  
// результат тренировки  
// первый подход  
// совершено 12 повторов (count)  
// вес (weight) снаряда – 12 кг  
  
// второй подход  
  
// упражнение активно (Deleted - удалено)  
// комментарий к упражнению  
  
// началось следующее упражнение на ту же дату  
  
// комментарий к упражнению  
  
// тут уже расписание на другую дату  
// начало упражнения – оно одно в этот день

## 5. Отправка данных о тренировках

### POST /api/v3/train

#### Input:

user\_id – идентификатор пользователя

token – токен пользователя

train – тот же формат, что и в поле data при GET /api/v3/train (т.е. подобно train === data)

#### Output:

```
data: {  
    <timestamp>: <Exercise>[]           // дата тренировки: массив упражнений  
}
```

**Примечание:** как мы видим –url АПИ в данном случае одинаков, отличается только метод – GET\POST.

GET- только отдает данные по тренировке

POST и отдает и принимает для сохранения.

Формат входных данных и выходных данных в случае метода POST одинаков для полей train и data.

Если в упражнении, в поле status, приходит значение “Deleted” – то упражнение будет удалено в расписании пользователя. Однако, даже удаляемое упражнение должно сохранить формат данных при передаче на сервер.

Если есть данные по времени отдыха\работы – то к параметрам w/c/d/t добавляем параметры tr/tw.

Если данных по времени нет – параметры tr/tw можно не добавлять в результаты тренировки.

**Нюанс:** единственный параметр, который можно не передавать в поле train на вход в данный метод – это поле comment, так как комментарии к упражнению только для чтения пользователем, то сервер не будет их сохранять.

## 6. Получение данных о записях

GET /api/v3/note

### Input:

user\_id – идентификатор пользователя  
token – токен пользователя

### Output:

```
data: {  
  <timestamp>: {  
    date:<string>  
    date_unix: <timestamp>  
    note:<string>  
  },  
  ...  
}
```

### Пример:

```
data: {  
  "1369252800": {  
    // дата в timestamp  
    "date": "2013-05-23",    // дата записи  
    "date_unix": "1369252800", // дата записи в timestamp  
    "note": "Test1<br>test1"  // сама запись  
  },  
  "1369339200": {  
    "date": "2013-05-24",  
    "date_unix": "1369339200",  
    "note": "Test2<br><br>test2"  
  }  
}
```

## 7. Отправка данных о записях

### POST /api/v3/note

#### Input:

user\_id – идентификатор пользователя

token – токен пользователя

note – тот же формат, что и в поле data при GET /api/v3/note (т.е. подобно note === data)

#### Output:

```
data: {  
  <timestamp>: {  
    date:<string>  
    date_unix: <timestamp>  
    note:<string>  
  },  
  ...  
}
```

**Примечание:** как мы видим –url АПИ в данном случае одинаков, отличается только метод – GET\POST.

GET- только отдает данные по записям

POST и отдает и принимает для сохранения.

Формат входных данных и выходных данных в случае метода POST одинаков для полей note и data.



## Примечание

### Типы объектов при ответе:

#### <Exercise>:

```
{
    date: <string>           // дата формата yyyy-mm-dd hh:mm:ss
    date_unix: <timestamp>    // тоже сама дата в timestamp
    id_exercise: <int>        // идентификатор упражнения
    id_muscle_group: <int>    // идентификатор мышечной группы
    mode: <string>            // степень нагрузки - Normal\Hard\Medium\Easy
    position: <int>           // номер позиции упражнения в текущем дне
    result: <b>Result</b>      // результат выполнения
    status:<string>           // статус – Active\Deleted
    comment: <string>        // комментарий к тренировке
}
```

#### <Result >:

```
{
    <int>: {                  // номер подхода в упражнении
        <char>:<float>        // параметр (w/c/d/t) и его значение
        ...                  // параметры указываются по порядку
    }
    ...
}

// подходов может быть несколько
```

#### <Units>:

```
{
    <string>: {              // группа применения
        <int>: {             // идентификатор внутри группы
            id:<int>          // идентификатор группы применения
            sym:<string>       // символ единиц измерений или сокращенное название
            name:<string>      // название для пользователя - русифицированное
            color:<string>     // цвет фона, пример - #FC9B00
            coeff:<int>        // коэффициент перевода по отношению к основной единице
        }
    }
}
```

## Переводы единиц измерения

На данный момент, при запросе общих данных пользователя (**/api/v3/syncdata**), в поле units придут следующие данные:

```
units: {  
  weight : {                                     // для работы с весом в упражнениях (к примеру, вес гантели)  
    1 : {  
      id : 1,  
      sym : кг,  
      name : килограммы,  
      color : #D14836,  
      coeff : 1,  
    },  
    2 : {  
      id : 2,  
      sym : lb,  
      name : фунты,  
      color : #3584CE,  
      coeff : 2.20462262,  
    },  
  },  
  distance : {                                   // для работы с расстояниями (бег\плавание и все что имеет тип d)  
    1 : {  
      id : 1,  
      sym : мет,  
      name : метры,  
      color : #D14836,  
      coeff : 1,  
    },  
    3 : {  
      id : 3,  
      sym : mil,  
      name : мили,  
      color : #3584CE,  
      coeff : 0.000621371192,  
    },  
  },  
  length : {                                     // для работы с измерениями тела – в приложении не используется  
    1 : {  
      id : 1,  
      sym : см,  
      name : сантиметры,  
      color : #D14836,  
      coeff : 1,  
    },  
    4 : {  
      id : 4,  
      sym : inch,  
      name : дюймы,  
      color : #00B760,  
      coeff : 0.393700787,  
    },  
    5 : {  
      id : 5,  
      sym : ft,  
      name : футы,  
      color : #FC9B00,  
      coeff : 0.032808399,  
    },  
  },  
};
```

Из ответа нам нужны группы weight и distance. В каждой группе есть единица измерения с идентификатором 1 (coeff также равен 1) – это главная единица измерения (по умолчанию так сказать), относительно которой рассчитываются коэффициенты для других единиц измерения. К примеру, чтобы перевести килограммы в фунты, надо килограммы умножить на coeff фунтов(2.20462262).

Однако, так как у нас единицы измерений можно менять последовательно, то общая формула расчета данных такая:

$$val^y = (val^x / coeff^x) * coeff^y$$

, где **y** – новая система измерений, а **x** – предыдущая система измерений.

Пример с измерением тела (для большей наглядности, так как там 3 единицы) – переведем по схеме сантиметры -> дюймы -> футы -> сантиметры:

1. Получаем 23 сантиметра
2. Переводим 23 сантиметра в дюймы:  
 **$(23 / 1) * 0.393700787 = 9.055118101$**  дюймов
3. Получившиеся 9.055118101 дюймов переводим в футы:  
 **$(9.055118101 / 0.393700787) * 0.032808399 = 0.754593177$**  футов
4. Получившиеся 0.754593177 футов переводим в сантиметры:  
 **$(0.754593177 / 0.032808399) * 1 = 23$**  сантиметра

В приложении необходимо сохранять достаточную точность перевода данных, чтобы многократная смена единиц измерения не повлияла на сам результат. Проще говоря – если я ввел 23 сантиметра и 100 раз поменял единицы измерения – в итоге я все равно должен получить ровно 23 сантиметра. Реализацию такой схемы можно посмотреть на сайте [fitmus.com](http://fitmus.com)

## Коды ошибок и их расшифровка

### **2х-3х ошибки - отсутствуют необходимые входные параметры или они пустые**

- 20 - не найден или пустой app\_id
- 21 - не найден или пустой app\_sign
- 22 - не найден или пустой data\_sign
- 25 - не найден или пустой логин
- 26 - Логин\email не правильный
- 27 - не найден или пустой user\_id
- 28 - user\_id не правильный
- 29 - подпись пользователя token - не найдена

### **5х ошибки авторизации**

- 50 app\_id не верный
- 51 app\_sign не верная
- 52 data\_sign не верный
- 53 token не верный

### **9х - ошибки БД**

- 91 - ошибка при записи данных в БД. Запрос прилагается.

### **10х - 11х неверные данные на сервер**

- 100 - не верный формат данных, общая(базовая) ошибка
- 101 - неправильный тип результата в пользовательском упражнении
- 102 - неправильный формат результата для сохранения - найден пустой массив
- 103 - неправильный формат результата для сохранения - не найдена валидная дата в одном из массивов
- 104 - неправильный формат результата для сохранения - не найден валидный идентификатор упражнения
- 105 - неправильный формат результата для сохранения - не найден валидный идентификатор группы мышц
- 106 - неправильный формат результата для сохранения - не найден валидный номер позиции упражнения (position)
- 107 - неправильный формат результата для сохранения - не найден валидный номер нагрузки (mode)
- 108 - неправильный формат результата для сохранения - не найден статус упражнения
- 109 - неправильный формат результата для сохранения - статус не верен
- 110 - неправильный формат результата для сохранения - не найдено поле с результатами упражнения (result)