

Соглашения по эргономике веб-приложений

[Введение](#)

[Термины](#)

[Принципы](#)

[Принцип симметрии](#)

[Принцип явного действия](#)

[Принцип обратной связи](#)

[Принцип чёткого местоположения \("где я?"\)](#)

[Принцип логической полноты \("сказал А, говори Б"\)](#)

[Соглашения для Rails-приложений](#)

[На всех `/show` должна быть нарисованная на экране кнопка возврата назад](#)

[В формах редактирования поля должны содержать комментарии и примеры](#)

[Порядок следования колонок в информационных таблицах \(например, `aindex``\)](#)

[Правила](#)

[\(Недописанное/размышления/мусор\)](#)

[Принцип классификации?](#)

Введение

Термины

- **надо, должен, обязательно, строго, всегда** -- необходимо **безусловно** следовать тому, что написано.
- **нельзя, запрещено, недопустимо, неправильно** -- следование тому, что нельзя, считается **ошибкой** и вас попросят это исправить.
- **допустимо, желательно, рекомендуется, нежелательно, не рекомендуется** -- принятие решения в конкретной ситуации зависит от вас, **следуйте голосу разума**.

Принципы

Принцип симметрии

1. Если можно зайти, то можно и выйти.
2. Если можно создать, то можно и удалить.
3. Если можно сделать действие, то можно и отказаться от действия.
4. Если есть средство для движения "вперёд" ("внутрь"), то должно быть и **аналогичное по форме** средство для движения "назад" ("наружу").

Неправильно:

1. Нажали "Выбрать пользователя" и перешли на страницу "Выберите пользователя", а вернуться (отказавшись от выбора) можем **только по Back** в браузере.
2. Нажали "Редактировать" и перешли на страницу "Редактирование пользователя", а **отказаться не можем** и вернуться можем только по Back в браузере.
3. В динамическом интерфейсе дважды ткнули в значение поле "Имя", чтобы его отредактировать. Всплыл динамический диалог редактирования, **в котором нет кнопки "Отмена"**.
4. В динамически всплывающем диалоге можно подтвердить ввод нажатием клавиши **Enter**, однако нельзя от него отказаться нажатием **клавиши Esc**.

Принцип явного действия

Принцип: Каждое из доступных действий по модификации данных должно быть оформлено в интерфейсе как "важное". Значимые данные должны модифицироваться **только в том случае**, если пользователь явно об этом "попросил", нажав на соответствующие "важные" кнопки.

Неправильно:

1. Выбрали в списке запись и нажали "Просмотреть". Это привело к тому, что в записи неявно изменилось одно или несколько значимых полей.

Неоднозначный случай, но в целом **неправильно** (лучше избегать):

2. Выбрали в списке запись и нажали "Редактировать". Ничего не меняя в полях, нажали "Сохранить". Это привело к тому, что в записи изменилось одно или несколько значимых полей.

Принцип обратной связи

Принцип: Каждое выполненное действие по модификации данных должно иметь **видимый эффект** (обратную связь). Каждое выполненное действие по изменению **представления данных** также должно иметь видимый эффект.

Неправильно:

1. Из списка пользователей зашли на страницу привязки признаков к пользователю. Отредактировали, вернулись, а запись в списке **осталась без видимых изменений**.
2. В динамическом интерфейсе ткнули в иконку "Редактировать имя". Всплыл диалог редактирования. Модифицировали имя и подтвердили ввод. Диалог растворился, а отображаемое имя **осталось без изменений**.
3. Отфильтровали список пользователей по полю "Признак", введя туда буквы "прогр". Список уменьшился со 100 записей до 4, однако **интерфейс не отображает**, какие именно признаки совпали.

Принцип чёткого местоположения ("где я?")

Средства навигации по сайту должны быть такими, чтобы у пользователя не возникало ощущения, что его "вдруг перекинули" неизвестно куда. Если некое пошаговое действие предполагает движение "вперёд", то возврат должен осуществляться именно "назад", а не куда попало.

Принцип: Приложение должно **в полном объёме** сохранять информационный контекст, чтобы поддерживать у пользователя **устойчивое ощущение**, что он вернулся в то же место, откуда пришёл.

Неправильно:

1. Отфильтровали список пользователей по фамилии, содержащей буквы "иван". Из 100 записей получили 4. Нашли нужного нам Иванова, ткнули "Редактировать". Отредактировали, нажали "Сохранить" и **оказались в списке пользователей, который не отфильтрован**.
2. Листаем список пользователей. На 5-й странице нашли нужного нам, ткнули "Редактировать". Отредактировали, нажали "Сохранить" и **оказались опять на 1-й странице списка**.

Принцип логической полноты ("сказал А, говори Б")

Наше приложение даёт пользователю возможность **вводить и анализировать данные**.

Всё, что отображается на экране -- производная от этих данных.

Принцип: Если отображается что-то, что по замыслу можно редактировать, должна быть предоставлена возможность это редактировать.

Неправильно:

1. В списке пользователей отображаются поля "Имя" и "Пол". В форме редактирования пользователя мы можем модифицировать имя, **но не можем** модифицировать пол.
2. В списке пользователей отображается фотография и напротив записи админа она есть. Однако **у остальных её нет и не предоставлено никаких средств**, чтобы её загрузить.

Соглашения для Rails-приложений

На всех `/show` должна быть нарисованная на экране кнопка возврата назад

Данное требование является прямым следствием принципа симметрии и принципа чёткого местоположения.

Пример из ПроСтора:

```
<%= render partial: "navigate_back" %>
```

В формах редактирования поля должны содержать комментарии и примеры

Как тут:

```
<div class="Comment">Имя файла с расширением, например, <tt>Логотип  
5.psd</tt>.</div>  
<div class="Comment">Например, <tt>Старлей</tt>.</div>
```

Порядок следования колонок в информационных таблицах (например, `aindex`)

Правила

1. **Всегда делаем по аналогии.** Ищем похожий пример в недавно сделанном,

подмечаем важное и делаем как там.

2. **Более важные** колонки выводим первыми (**левее**), **менее важные** -- последними (**правее**).
3. Содержимое маловажных колонок **размечаем соответствующим стилем**, чтобы за них меньше цеплялся глаз.
4. Стараемся группировать колонки по смыслу так, чтобы **логически связанные значения находились рядом**.

(Недописанное/размышления/мусор)

Принцип классификации?

Тут что-то вроде того, что все кнопки должны быть жёлтыми, всё что можно редактировать должно содержать элементы оформления А, а что нельзя -- элементы оформления Б. До конца не сформулировано, тут огрызки.

1. Если отображается что-то, что в **данном случае** редактировать запрещено (нет прав и пр.), должно внятно визуально отображаться, что именно в данном случае редактировать запрещено.
2. Если отображается что-то, что редактировать невозможно (например, количество записей типа N), должны присутствовать внятные визуальные намёки, что **такие вот значения** редактировать нельзя.

Неправильно:

1. В списке Пользователей отображаются поля "Имя" и "nПроектов". С помощью динамического интерфейса можно двойным тычком в значение поля "Имя" редактировать его. Значение поля "nПроектов" является счётчиком, поэтому редактировать его нельзя. Однако визуально поля "Имя" и "nПроектов" оформлены абсолютно одинаково.