

Компоновщик отображения

05-03-2011

Содержание.

[1. Макет и общие разъяснения.](#)

[2. Основные компоненты.](#)

[3. Функциональность.](#)

[4. Модель данных.](#)

[5. Соглашения об именах и каталогах.](#)

[6. Используемые готовые и полу-готовые решения и библиотеки.](#)

[7. Стилизация.](#)

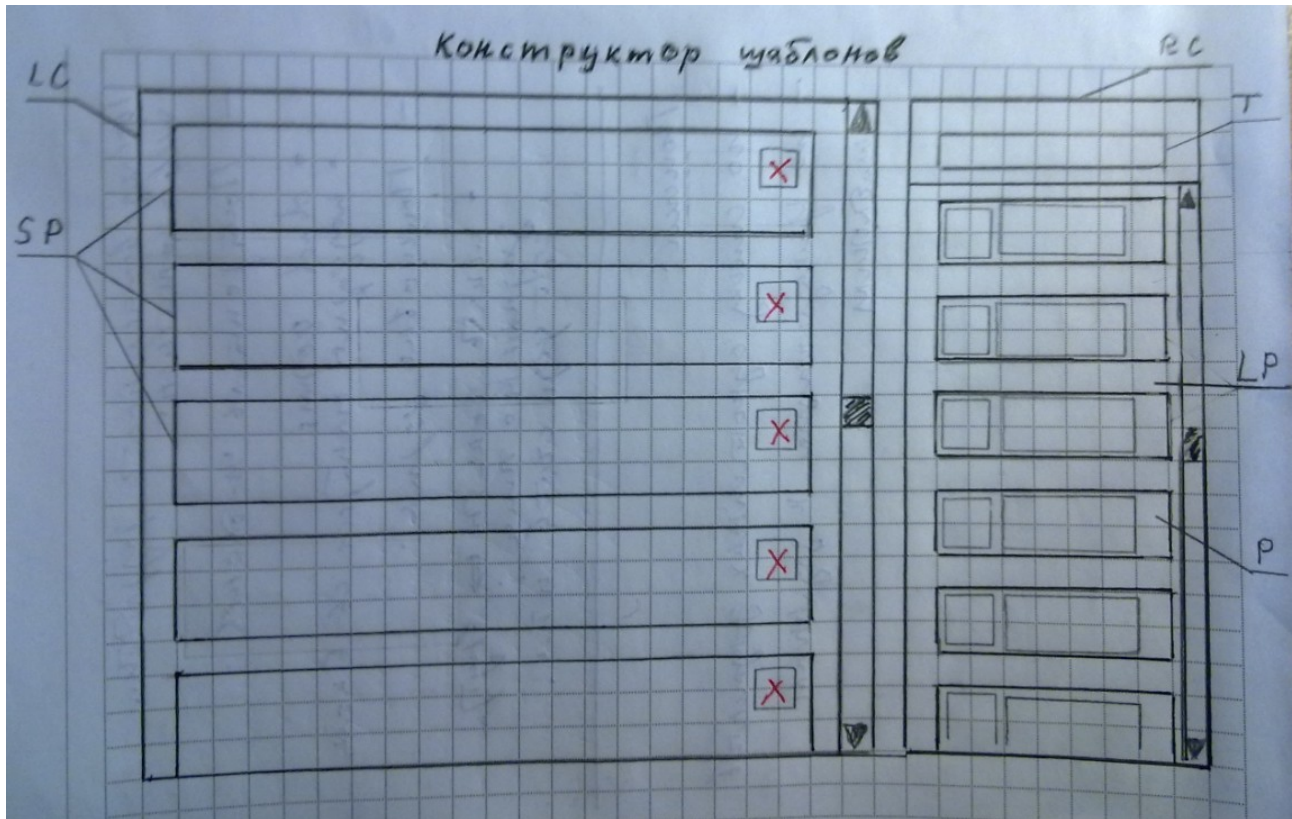
[8. Требуемый результат.](#)

[9. Сроки.](#)

[10. Рабочие коммуникации.](#)

1. Макет и общие разъяснения.

Компоновка аналогично Конструктору шаблонов:



Отличие состоит в способе переноса объектов из правого контейнера в левый и в свойствах объектов, размещаемых в левом контейнере.

Смысловое назначение Компоновщика отображения следующее:

есть набор форм (объектов), символически представленных в правом контейнере в виде блоков с иконками и пояснительными текстами. Пользователь знает, что за формы там реально представлены и хочет удобным для себя образом расположить их в рабочей области (левый контейнер). Он выбирает и перемещает мышкой выбранный объект в левый контейнер и там происходит дроп с выводом уже реальной формы, соответствующей выбранному справа. Форму в левом контейнере пользователь может перемещать мышкой и изменять ее размер. В итоге получается желаемая визуальная конфигурация, которая сохраняется в базе. Далее (в других местах программы) при отображении используются сохраненные данные чтобы динамически разместить нужные компоненты и установить им заданные размеры.

2. Основные компоненты.

Аналогично Контейнеру строителя шаблонов.

3. Функциональность.

Объекты из правой панели перемещаются мышкой в левую панель. При этом в левой панели выкладывается код, отличный от кода объекта, выбранного в правой панели. Иными словами объекты в правой панели нужны для обозначения того, что выбирается, тогда как при выкладывании в левую панель помещается код, соответствующий выбранному объекту. Например в правой панели может быть иконка, обозначающая форму и при выкладывании в левую панель помещается не иконка а форма.

Объекты, помещенные в левой панели обладают свойствами:

- Изменения размера мышкой
- Перемещения в пределах левой панели

При этом левая панель также обладает свойством изменения размера по вертикали для возможности размещения в ней выбранных объектов.

Объект на левой панели может быть удален аналогично тому как это делается в контейнере строителя шаблонов.

В качестве объекта размещаемого в левой панели сделать блок **div** заполненный текстом. Если при изменении размера текст перестает уместиться, то автоматически включается **scroll** в **div**. Порядок размещения текстового блока в два этапа: сначала на левой панели в месте где произведен дроп создается блок **div** стандартного размера (размер определить визуалью на свой вкус но не более **75%** по ширине и не более **15 - 20%** по высоте) и затем в него подгружается страница с текстом. Создать набор таких простых текстовых страниц с именами **1.HTML 2.HTML** и тд где имя страницы есть целочисленный **id** объекта правой панели. Так если выбран на правой панели объект **cid 12**, то в левой панели подгружаемая страница **12.html**.

В верхней части правой панели где отведено место для заголовка добавить иконки:

Очистить все. При клике на нее выдается **jquery dialog** с вопросом на подтверждение и удаляются все объекты на левой панели.

Иконка 'з агрузить'. При клике на нее очищается левая панель и производится ее заполнение по данным, записанным в заданном массиве **json** объектов. Это нужно для отработки динамической загрузки и компоновки объектов на левой панели по данным сохраненным из предыдущих построений. Для этого создать в коде заранее готовый массив **json** объектов который как бы загружен уже с сервера и по нему произвести заполнение левой панели.

Иконки подобрать из галереи **famfamfam**.

4. Модель данных.

В модели данных должна храниться информация о выбранных объектах, т.е. Объектах находящихся в левой панели, а также информация об их размерах и положении.

Для хранения информации используется массив объектов `json`, каждый из которых содержит информацию о конкретном объекте из левой панели.

Структура `json` объекта:

`ID` -> значение `ID` (как в контейнере строителя шаблонов)

`W` -> ширина

`H` -> высота

`X` -> координата по горизонтали

`Y` -> координата по вертикали

При добавлении объекта в левую панель, при изменении его размера или положения соответственно изменяются параметры `vjson`.

При удалении объекта из левой панели описывающий его `json` удаляется из массива. Массив сжимается чтобы не образовывались `null` значения.

5. Соглашения об именах и каталогах.

Прикладные html-файлы (не относящиеся к включаемым библиотекам, плагинам и т.д.) размещаются в корневом каталоге разработки.

Прикладные js-файлы размещаются в каталоге “js”.

Прикладные css-файлы размещаются в каталоге “css”.

Все файлы, относящиеся к jQuery, размещаются в каталоге “jq” и его необходимых подкаталогах.

Прикладные файлы именуются по принципу:

filename.html

filenameDR.js - прикладной скрипт, включаемый в страницу и, как правило, обрабатывающий jquery-событие “documentready”.

6. Используемые готовые и полу-готовые решения и библиотеки.

В разработке использовать методы и функции библиотеки jquery 1.3.2. Только там, где это по каким-либо причинам невозможно, использовать native java script.

Подключение функциональности загружаемой страницы происходит через включаемый в загружаемую страницу ее собственный именной скрипт и прописанный в нем метод `$document).ready`. (см.Соглашения об именах).

В кодах, для единообразия стиля программирования, использовать примеры из demo bundle, поставляемый от jquery (www.jquery.com).

При работе с готовыми компонентами (tabs, menu, accordion, table etc) использовать встроенные в них методы, настройки, обработчики событий и т.д. Не изобретать велосипед!

Для симуляции обращения к серверу использовать свою любую фэйковую страницу на сервере. Вызов производить только через `$.get` or `$.post` or `$.getJSON` or `$.getScript`.

Для синхронизации загрузок использовать функции обратного вызова jquery. Не переполнять ajax pool браузера!

Для загрузки вложенных страниц использовать jquery метод `load`.

7. Стилизация.

Использовать как образец

<http://www.basev.ru/client/development1/mainform/index.html>

<http://www.basev.ru/client/development1/linksnavigator/index.html>

<http://www.basev.ru/client/development1/patternbuildercontainer/index.html>

Иконки брать из галереи famfamfam (см. рабочий каталог).

8. Требуемый результат.

Результатом работы является набор html-страниц, js- и css-файлов, размещенных на сервере разработки в заданном каталоге. Страницы должны отвечать описанным выше требованиям.

Код скриптов должен быть отформатирован, функции и существенные переменные должны быть подписаны рядом в комментариях. В заголовке скрипта разместить главный комментарий с кратким описанием как работает страница с указанием существенных функций и переменных.

Код страниц должен быть отформатирован. Основные элементы должны быть подписаны рядом в комментариях. В том числе указывать в комментариях закрытия какой именно раздел открывается и закрывается (для <div> и </div>) для существенных разделов.

Модель данных в коде (объявления и существенные места использования переменных) должны сопровождаться комментариями.

<ftp://ftp.basev.nichost.ru>

Имя пользователя (логин): basev_developer3
пароль jyl40hg7

9. Сроки.

На разработку отводится 3 рабочих дня с поставкой не позже, чем на 5-ый календарный день с начала работы. Возможно добавление 1 рабочего дня на уточнения и подгонку.

10. Рабочие коммуникации.

Основное взаимодействие происходит по электронной почте на адрес руководителя разработки Уланова Александра: oulanov@right-bracket.com или запасной адрес a.v.oulanov@gmail.com . Для срочной связи использовать телефон (голосовая связь, смс, ммс):

+ 07 916 827 2607

Контакты с руководителем организации Березой Константином по адресу электронной почты bereza@right-bracket.com

Телефонные контакты - по предварительной договоренности по электронной почте.